

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
 - TEXT CUT OFF AT TOP, BOTTOM OR SIDES
 - FADED TEXT
 - ILLEGIBLE TEXT
 - SKEWED/SLANTED IMAGES
 - COLORED PHOTOS
 - BLACK OR VERY BLACK AND WHITE DARK PHOTOS
-
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

THIS PAGE BLANK (USPTO)



Europäisches
Patentamt

European
Patent Office

Office eur péen
des brevets

PCT/GB 00 / 00980

09/914289

GB 00/980

REC'D 24 MAY 2000

WIPO

PCT

Bescheinigung

Certificate

Attestation

Die angehefteten Unterla-
gen stimmen mit der
ursprünglich eingereichten
Fassung der auf dem näch-
sten Blatt bezeichneten
europäischen Patentanmel-
dung überein.

The attached documents
are exact copies of the
European patent application
described on the following
page, as originally filed.

Les documents fixés à
cette attestation sont
conformes à la version
initialement déposée de
la demande de brevet
européen spécifiée à la
page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

99305539.1

**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 05/04/00
LA HAYE, LE

THIS PAGE BLANK (USPTO)



Europäisches
Patentamt

European
Patent Office

Office eur péen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99305539.1

Anmeldetag:
Date of filing: 20/12/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
BRITISH TELECOMMUNICATIONS public limited company
London EC1A 7AJ
UNITED KINGDOM

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Computer system

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du dépôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

COMPUTER SYSTEM

This invention relates to a computer system.

As is well known, a computer system may include an application program
5 installed and loaded on to the computer system and which is capable of accessing
one or more data sets stored on the computing system. As is well known, an
application program may be updated from time to time to produce a new version with
additional functionality. Thus, it is quite usual to produce a series of versions of a
particular application program. The structure of a data set may also be updated from
10 time to time, for example by adding one or more additional fields. With presently
known computing systems, when the structure of a data set is updated, it is
necessary to produce an updated version of the associated application program in
order to enable it to access the data set. Some users of the application program may
not wish to suffer the inconvenience of obtaining an updated version. Also, after a
15 new version of an application program has been produced, the users of it will have to
be trained how to use it and so there will be a period during which some users will
want to continue using the previous version of the application program.

According to one aspect of this invention there is provided a computer
system including a version of an application program stored in said computer system,
20 at least one data access component stored in said computer system, the or each data
access component being associated with a set of data stored in said computer
system, the or each data access component being arranged to received a data query
from said version of said application program, access its associated data set in
response to said query, and transmit a reply to said version of said application
25 program, and one of said data access components being arranged to access its
associated data set both to respond to a query designed for the present version of the
structure of its associated set and also to respond to a query designed for an earlier
version of the structure of its associated data set.

In this specification, the term "data set" is intended to refer to a set of
30 functionally related data records which are accessed together as a unit.

With this invention, when the structure of a data set is updated, it is only
necessary initially to produce an updated version of the associated data access
component and not the associated application program. After th updated version of

the data access component has been installed on the computer system, the present version of the application program will be able to access the data set in the new structural version. Also, when an updated version of the application program is produced and installed on a computer, this will also be able to access the data set in
5 the new structural version.

The or each data set may comprise a set of records, each of which has a set of fields, and the difference between the present and earlier versions of the structure of the data set associated with said one data access component is that the records of the present version of the structure of the data set include at least one additional
10 field.

Preferably, each data access component comprises at least one software object, and said one data access component comprises at least two software objects, one of which is arranged to respond to a query designed for the present version of the structure of its associated data set, and the other of which is designed to respond
15 to a query designed for the earlier version of the structure of its associated data set.

The computer system may include a client computer, a server computer and a communications network arranged to connect the client computer and server computer, said version of said application program and said at least one data access component being stored on said client computer, and said data sets being stored on
20 said server computer.

There may be a plurality of versions of said application program stored on said server computer, said client computer being arranged to download one or more versions of said versions application program from said server computer for use on said client computer.

25 Said at least one data access component may be stored on said server computer, and said client computer may be arranged to download said at least one data access component from said server computer for use on said client computer.

When operating the computer system, preferably a version of the application program and said at least one data access component are downloaded in separate
30 operations.

According to another aspect of this invention there is provided a method of operating a computer system, said computer system including a version of an application program stored in said computer system, at least one data access

component stored in said computer system, the or each data access component being associated with a set of data stored in said computer system, the or each data access component being arranged to receive a data query from said version of said application program, access its associated data set in response to said query, and
5 transmit a reply to said version of said application program, said method comprising the steps of changing the structure of one of said data sets, and modifying the data access component associated with said one data set, the modification to said data access component permitting it to handle queries from the present version of the application program directed to said one data set but designed for the data set before
10 the change in its structure.

This invention will now be described in more detail, by way of example, with reference to the drawings in which:

Figure 1 is a block diagram of a directory assistance system for a telecommunications network embodying this invention;

15 Figure 2 shows the structure of the records of a data set used by the system of Figure 1;

Figure 3 shows an updated version of the structure for the records shown in Figure 2;

Figure 4 shows the software components of a workstation forming part of
20 the directory assistance system of Figure 1;

Figure 5 is a flow chart illustrating the operation of the workstation;

each of Figures 6 to 10 is a functional block diagram showing an application program and associated data access components which may be installed in the workstation together with their interaction with data sets which may be stored in a
25 server computer forming part of the directory assistance system of Figure 1; and

Figure 11 is a block diagram illustrating the construction of a data access component.

Referring now to Figure 1, there is shown a directory assistance system 10 for a public switched telecommunications network (PSTN) 12. The directory
30 assistance system 10 includes a switch 14 connected to the PSTN 12 and which distributes incoming calls to a set of workstations used by directory assistance operators. One of these workstations is shown and indicated by reference numeral 16. The workstations, including workstation 16, are connected to a local area

network (LAN) 18. LAN 18 is connected to a local server computer 20, the purpose of which will be described in more detail below. LAN 18 is also connected to a router 21 which in turn is connected to a wide area network (WAN) 22. WAN 22 is connected to a remote server computer 24. The remote server computer 24 stores a
5 large set of records, each of which contains details of one of the customers of the PSTN 12. In each record, the details include the name, address and telephone number of the customer.

The workstation 16 and other workstations connected to the switch 14, the LAN 18, the local server computer 20 and the router 21 are co-located in the same
10 physical site. The remote server computer 24 also serves workstations located at other sites.

Each of the workstations, including workstation 16, has an application program which can perform a search on the records stored in the remote server computer 24 to retrieve the telephone number of a customer of PSTN 12 in response
15 to an enquiry from a caller. When performing a search, the user or operator enters the name of the customer on the workstation 16 and usually other details such as the initials of the customer, the town where the customer is located or part or all of the address of the customer. As will be described below, the user can also be assisted by accessing records stored in the local server 20.

20 From time to time, the format or structure of the records stored in the remote server 24 may be updated to produce a new version. When a new version is produced, the application program used by the workstations must also be updated so as to enable it to access the new version. For a transitional period, the old and new versions of the records may both be stored on the remote server 24 or one version
25 may be stored on the remote server 24 and another version stored on another server computer connected to WAN 22. This enables the records to be accessed using both the old and new versions of the application program.

As may be readily appreciated, the directory assistance system 10 is also a computing system. The computers include the workstations, the local server
30 computer 20 and the remote server computer 24. The hardware construction of each of these computers is conventional and includes a central processing unit, a store formed from a combination of read-only-memory, random-access-memory, hard disk storage and floppy disk storage, a keyboard, visual display unit and input and output

ports. In each of the computers, an operating system and one or more application programs are held in the store and provide the desired functionality.

The various versions of the application program used by the workstation 16 are stored in the local server 20. As will be described in more detail below, each workstation is arranged to download the various versions of the application program from the local server 20. For each user of the workstations, the local server 20 contains a record setting out the details of the user and the version or versions of the application program which that user is permitted to use. Several sets of data records are also stored on the local server 20. Each of these sets of records is designed to assist the users of the workstations in answering enquiry calls.

One of these sets of data is a set of locality records. When making an enquiry for the telephone number of a customer of the PSTN 12, a caller will sometimes give incorrect information with regard to the locality of the customer. For example, rather than giving the correct town in which the customer is located, the caller may give the name of a nearby town. The purpose of the locality records is to enable a user of a workstation to extend the search to neighbouring areas. The fields of a typical locality record are shown in Figure 2. As may be seen, these records include a field 40 for a town, a field 41 for the nearest town to the town entered in field 40, a field 42 for the next nearest town and a field 42 for the county in which the town is located. Thus, when using this set of records, the user accesses the record for the town given by the caller. The user may then use the data in the record to extend the search to the nearest town, the next nearest town or to the county in which that town is located.

In the system of Figure 1, it may be desired to change the structural format of one of the sets of data records. For example, the structure of one of the sets of data records could be changed by adding an additional field. Referring to Figure 3, there is shown a modified structure for the set of locality records. In the structure shown in Figure 3, each locality record has a field 50 for a town, a field 51 for the nearest town, a field 52 for the next nearest town, a field 53 for the county in which the town is located and a field 54 for the nearest county. Thus, in comparison with the structure shown in Figure 2, the structure shown in Figure 3 includes an additional field, namely the field 54, for the nearest county.

When the structure of a set of data records is modified, for example as illustrated with reference to Figures 2 and 3, then the application program which is used to access those data records must be updated to take advantage of the modification. Also, in the absence of the present invention, the application program
5 would have to be updated even to access data records arranged in the new structure.

In this invention, and as will be described in more detail below, each set of data records stored in the local server computer 20 has an associated data access component. A copy of this data access component is loaded into each workstation which accesses those data records. In use, an application program installed on a
10 workstation does not access a set of data records stored in the local server computer 20 directly. Instead, it accesses the data records via the associated data access component.

With the invention, when the structure of a set of data records is changed, then the associated data access component is modified. The data access component
15 is modified so that both the existing version of the application program and the modified version of the application program will be able to access the data records with the modified structure via the modified data access component. When the existing version of the application program accesses the data records, it will not be able to take advantage of any extra features, for example an extra field, in the
20 modified structure of the data records. In contrast, the modified version of the application program will be able to take advantage of any additional features.

Thus, with the invention, when the structure of a set of data records is modified, it is necessary initially only to modify the data access component and not the whole application program itself. The invention also provides the advantage that
25 the data records for the new structure can be used with both the existing and modified versions of the application program. This allow the users of the application program to continue using the data records while they are being trained in the use of the modified version of the application program.

Referring now to Figure 4, there are shown the main software components
30 installed into each of the workstations, such as the workstation 16, of the directory assistance system 10. As shown in Figure 4, these components include an operating system 60, a control program 61, application programs 62 and a set of data access components 63. Normally, a complete set of the current versions of the main

application program are loaded into each of the workstations. The main function of the control program 61 is to download the current versions from the local server computer 20 and to delete any version which is no longer in use. As explained above, each data access component is associated with a set of data records stored on the local server 20.

Referring now to Figure 5, there is shown a flow chart illustrating the operation of one of the workstations.

Initially, in a step 70, the control program obtains a list of available versions of the main application program from the server computer. The main application program is the program which is used for answering enquiries from calling parties. All further references in this description to the application program are to be understood as references to this main application program.

Next, in a step 72, the control program obtains a list of the versions of the application program installed on the workstation. Then, in a step 73, any version of the application program which is no longer available is removed by deleting from the store of the workstation.

Next, in a step 74, the control program downloads from the server computer any available versions of the application program which are not currently installed on the workstation. Then, in a step 75, the current versions of the data access components are downloaded from the server computer and stored in the workstation. These current versions replace any previous versions stored in the workstation.

In a step 76, the control program starts-up the last used version of the application program. The last used version of application program is the version of the application which has been most recently used on the workstation.

The application program itself, in a step 77, then obtains the data record for the present user of the workstation from the server computer. The user record specifies the version or versions which the user is permitted to use. In a step 78, the user then selects the required version of the application program. If the user is permitted to use more than one version of the application program, the user makes a selection based on the user's present requirements. For example, the user may use one version for answering enquiries and another version for training.

In a step 79, the application program checks whether the required version is running. If the required version is running, a jump is made to step 80. If the required

version is not running, in steps 81 and 82, the control program shuts down the version of the application program which is currently running and then starts up the required version. This is followed by step 80. In step 80, the user uses the application program either for training or for answering enquiries from calling parties.

5 Referring now to Figure 6, there is shown the arrangement of version 1.0 of the application program, two data access components dmc1 and dmc2 and two data sets A and B. The data access components dmc1 and dmc2 provide access to the data sets A and B, respectively, as indicated by arrows 90 and 91. The interfaces between the application program and data management components dmc1 and dmc2
10 are provided, respectively, by interfaces iDmc1 and iDmc2. These interfaces are provided by portions of code in the application program and the data management components.

Each of the data management components dmc1 and dmc2 is constructed so as to provide server functionality and each of these data management components is
15 arranged to respond to queries from the application program. Thus, for example, when the data access component dmc1 receives a query from the application program across the interface iDmc1, the data access component dmc1 accesses the data set A to obtain the required information. After obtaining the required information, the data access component dmc1 sends a response across the interface
20 iDmc1 to the application program. The data sets A and B are, as explained above, stored in the local server computer 20.

Referring now to Figure 7, there is shown a modification to the arrangement of Figure 6. In this modification, the data set A has been replaced by a modified data set A_v2 and the data access component dmc1 has been replaced by a modified data
25 access component dmc1_v2. In comparison with the data set A, the structure of the data set A_v2 has been modified, for example by adding an extra field to each record. The modified data access component dmc1_v2 is arranged to access the modified data set A_v2. However, the data access component dmc1_v2 presents the same interface iDmc1 to the application program as that presented by the original data
30 access component dmc1. Thus, the modified data access component dmc1_v2 can respond to queries received across the interface iDmc1. However, as the application program is still version 1.0 of the application program, it cannot take advantage of the extra functionality provided by the modification to the structure of the data set.

It should be noted that no change has been necessary to the application program itself and that only the data access component has been modified.

Referring now to Figures 8 and 9, there are shown two modifications to the arrangement of Figure 6. In each of these arrangements, the data set B has been replaced by a modified data set B_v2 and the data management component dmc2 has been replaced by a modified data management component dmc2_v2. In comparison with the data set B shown in Figure 6, the structure of the data set B_v2 has been modified, for example by adding an additional field. The arrangement of Figure 8 uses version 1.0 of the application program whereas the arrangement shown in Figure 9 uses version 2.0 of the application program. In comparison with version 1.0 of the application program, version 2.0 is designed to produce queries designed for the modified data set B_v2 and thus to take advantage of the additional functionality which is present in modified data set B_v2.

The modified data access component dmc2_v2 is designed to present the interface iDmc2 to version 1.0 of the application program and to present a modified interface iDmc2_2 to version 2.0 of the application program. For this reason, the interface iDmc2 is indicated by arrows joined by a solid line in Figure 8 but by arrows joined by a dotted line in Figure 9. Likewise, the modified interface iDmc2_2 is indicated by arrows joined by a dotted line in Figure 8 and a solid line in Figure 9.

When the modified data access component dmc2_v2 is used with version 1.0 of the application program, it receives and responds to queries over the interface iDmc2. When it is used with version 2.0 of the application program, it receives and responds to queries over the modified interface iDmc2_2.

As will be described with reference to Figure 10, it is possible to arrange data access components so that a single interface to an application program can be used for making queries to two or more data sets.

Referring now to Figure 10, there is shown an application program, two upper level data management components dmc3 and dmc4, two lower level data access components dmc3x and dmc3y and three data sets X, Y and Z. The data sets X, Y and Z are stored in the local server computer 20.

The data access components dmc3 and dmc4 present, respectively, interfaces iDmc3 and iDmc4 to the application program. The data access component

dmc4 is arranged to respond to queries from the application program received on the interface iDmc4 and directed to the data set Z.

The data access component dmc3 has an interface to each of the lower level data access components dmc3x and dmc3y. The lower level data access component
5 dmc3x is arranged to access data set X and the lower level data access component dmc3y is arranged to access the data set Y. Collectively, the data access components dmc3, dmc3x and dmc3y are arranged to respond to queries received on the interface iDmc3 and directed to either data set X or data set Y.

In a similar manner to that described with reference to Figures 6 to 9, if the
10 structure of one of the data sets X, Y and Z is modified, then the associated data access component is modified accordingly. Thus, for example, if the structure of data set X is modified, then data access component dmc3x is modified.

Each data access component is formed from a number of software objects. Specifically, each data access component includes an individual software object for
15 each interface provided by that data access component and a software object for accessing the associate data set.

Referring now to Figure 11, there are shown the software objects which form the data access component dmc2_v2. These software objects include two software objects CDmc2 and CDmc2_2 which, respectively, provide the interfaces iDmc2 and
20 iDmc2_2. The data access component dmc2_v2 also includes a software object CDsetB_v2 for accessing the data set B_v2. The software object CDsetB_v2 can receive messages from, and transmit messages to, each of the software object CDmc2 and CDmc2_2.

Each application program includes a set of software objects which generate
25 messages to, and receive messages from, the interface objects in the associated data access components. In relation to a data access component, each of the software objects functions as a client object. By way of example, Figure 10 shows a client software object ClientObj1 which can send messages to, and receive messages from, the interface object CDmc2 across the interface iDmc2.

30 In each data access component, the individual software objects collectively provide the methods which are needed for performing searches on the associated data set and for retrieving items of data for use by an application program. The data set associated with the data access component is private to the software objects of

that data access component in the sense that it can only be accessed by these software objects. The data set is thus encapsulated by the software objects of the associated data access component. The concept of encapsulation is well known in object oriented design.

- 5 In each data access component, the individual software objects may be formed from a group of subsidiary objects.

 Although this invention has been described with reference to a directory enquiry system, it is to be appreciated that it is suitable for general use in computing systems.

- 10 Although this invention has been described with reference to a computing systems formed from several computers, it is to appreciated that it is also suitable for use on a computing system formed from a single computer. Where the invention is used with a computing system formed from a single computer, then the application program or programs, data access component or components and the associated data
15 set or sets are all stored on the single computer.

THIS PAGE BLANK (USPTO)

CLAIMS

1. A computer system including:
a version of an application program stored in said computer system;
5 at least one data access component stored in said computer system, the or
each data access component being associated with a set of data stored in said
computer system;
the or each data access component being arranged to receive a data query
from said version of said application program, access its associated data set in
10 response to said query, and transmit a reply to said version of said application
program; and
one of said data access components being arranged to access its associated
data set both to respond to a query designed for the present version of the structure
of its associated data set and also to respond to a query designed for an earlier
15 version of the structure of its associated data set.
2. A computer system as claimed in claim 1, in which the or each data set
comprises a set of records each of which has a set of fields, and the difference
between the present and earlier versions of the structure of the data set associated
20 with said one data access component is that the records of the present version
include at least one additional field.
3. A computer system as claimed in any one of the preceding claims, in which
each data access component comprises at least one software object, and said one
25 data access component comprises at least two software objects, one of which is
arranged to respond to a query designed for the present version of the structure of its
associated data set and the other of which is arranged to respond to a query designed
for the earlier version of the structure of its associated data set.
- 30 4. A computer system as claimed in any one of the preceding claims, in which:
the computer system includes a client computer, a server computer and a
communications network arranged to connect the client computer to the server
computer; and

said version of said application program and said at least one data access component are stored on said client computer; and
said data sets are stored in said server computer.

- 5 5. A computer system as claimed in claim 4, in which a plurality of versions of said application program are stored on said server computer, and said client computer is arranged to download one or more of said versions of said application program from said server computer for use on said client computer.
- 10 6. A computer system as claimed in claim 5, in which said at least one data access component is stored on said server computer, and said client computer is arranged to download said at least one data access component from said server computer for use on said client computer.
- 15 7. A method of operating the computer system of claim 6, in which at least one version of the application program and said at least one data access component are downloaded from the server computer in separate operations.
8. A method of operating a computer system, said computer system including a
20 version of an application program stored in said computer system, at least one data access component stored in said computer system, the or each data access component being associated with a set of data stored in said computer system, the or each data access component being arranged to receive a data query from said version of said application program, access its associated data set in response to said query,
25 and transmit a reply to said version of said application program, said method comprising the steps of:
changing the structure of one of said data sets; and
modifying the data access component associated with said one data set;
the modification to said data access component permitting it to handle
30 queries from the present version of the application program directed to said one data set but designed for the data set before the change in its structure.

ABSTRACTCOMPUTING SYSTEM

5 A computing system includes an application program and two data access components dmc1 and dmc2 loaded on to a client computer. The data access components dmc1 and dmc2 provide access to data sets A and B stored on a server computer. Each data access component is arranged to handle a data query from th application program directed to its associated data set. If the structure of one of the data sets is changed, for example to add a new field, then the associated data access component is also modified. The modification to the data access component permits it to handle queries from the present version of the application program directed to the data set but designed for the data set before the change in its structure. The modification to the data access component also enables it to handle queries directed to the data set with the changed structure from an updated application program which is designed to take advantage of the new structure of the data set.

Figure (6)

THIS PAGE BLANK (USPTO)

1/11

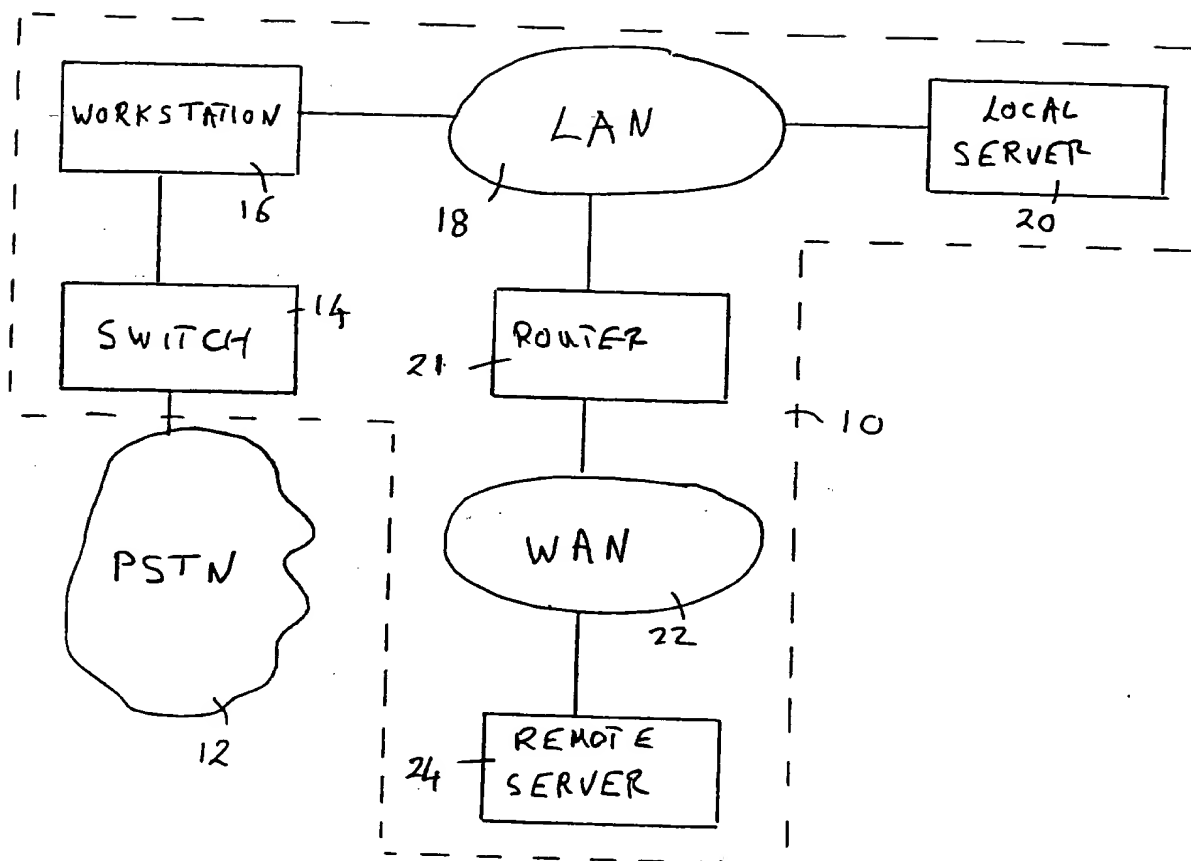


Fig. 1

2/11

TOWN	NEAREST TOWN	NEXT NEAREST TOWN	COUNTY
40	41	42	43

Fig. 2

TOWN	NEAREST TOWN	NEXT NEAREST TOWN	COUNTY	NEAREST COUNTY
50	51	52	53	54

Fig. 3

3/11

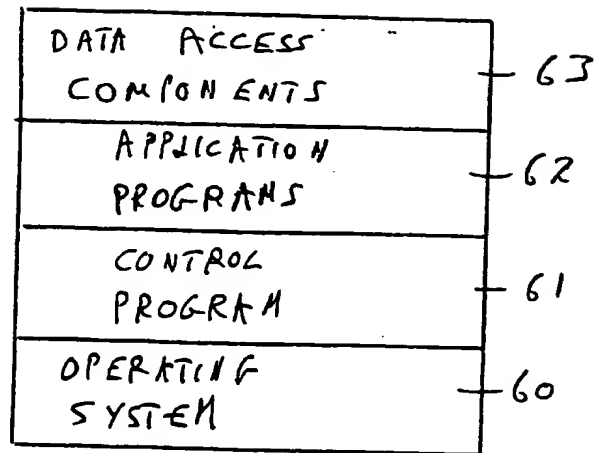


Fig. 4

THIS PAGE BLANK (USPTO)